7. Garold

Program Name: Garold.java **Input File: garold.dat**

You and your friend Garold have been playing a game called super tic tac toe. The main issue you have been having is that the board is somewhat confusing and you never know who's won. You need to write a program to take in a given super tic tac toe board and find out who has won. The rules of super tic tac toe are as follows:
Each super tic tac toe board will be made up of 9 regular tic tac toe boards, arranged with one regular board making up a cell. In order to win super tic tac toe, you need to win 3 of the regular boards in a row (so if you win the top 3 boards you win super tic tac toe). You can win if you get all of any column, row or either of the diagonals. To win each board, you need to win a row, column or diagonal, just like in regular tic tac toe. You need to output who wins the game.

Input: The input will begin with an integer, `n` (`0 < n <= 1000`), denoting the number of test cases to follow. Each test case will consist of 9 lines of 9 characters each, made up of `'X'` – denoting an X on the board, `'O'` – denoting an O on the board, `'.'` – denoting an empty space on the board. Every 3x3 characters of input denote a board (top left 3x3 denotes the top left board on the super tic tac toe board, and so on). Each 3x3 board will have a maximum of one winner, but there is no guarantee that the overall board is won by either team.

Output: If one of the players has won the game, output the string `"Player "`, followed by an `X` or an `O`, denoting which player won, followed by `"Won."`. If neither player has won the game, output `"Cat's Game."`. After outputting this string, output the the layout of the super board, with each 3x3 board denoted by an `X`, `O`, or `.`, denoting the winner if the board was won, or a `.` if no one has won the board. See the Sample output for more information.

Sample input:

```
2
X.OOOOXO.
OX..X.XX.
OOXX..OOX
X.O..X.OO
XOO.O.OO.
X..OO.OXX
XXXOOO.OX
OO..X.XO.
O.XXOXOOX
X.OOOOXO.
O...X.XX.
OOXX..OOX
X.O..X.OO
.OO.O.OO.
X..OO.OXX
X.OOO..OX
OO..X.XO.
O.XXOXXXX
```

Sample output:

```
Player X Won.
XOX
X.O
XOO
Cat's Game.
.OX
..O
O.X
```